OCT 23 2006

AF IFW

PATENT
Customer Number 22,852
Attorney Docket No. 07447.0061-00000
Xerox Ref. No.: D/A0A25

CERTIFICATE UNDER 37 CFR § 1.10 OF MAILING BY "EXPRESS MAIL"

EV758329616US	October 20, 2006
USPS Express Mail Label Number	Date of Deposit

I hereby certify that this correspondence is being deposited with the United States Postal Services "Express Mail Post Office to Addressee" service under 37 CFR §1.10 on the date indicated above and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

By: Cindy Baglietto
Cindy Baglietto

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	
Eric GAUSSIER et al.	
Application No.: 09/982,236	Group Art Unit: 2161
Filed: October 19, 2001	Examiner: NGUYEN, Cam Linh T.
For: METHODS, SYSTEMS AND ARTICLES OF MANUFACTURE FOR SOFT HIERARCHICAL CLUSTERING OF CO-OCCURRING OBJECTS	Confirmation No.: 7611

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

TRANSMITTAL OF SUPPLEMENTAL APPEAL BRIEF (37 C.F.R. 41.37)

Transmitted herewith is the SUPPLEMENTAL APPEAL BRIEF in this application with respect to the Office Action mailed on July 20, 2006 and the Notice of Appeal originally filed on November 22, 2005.

This application is on behalf of

☐ Small Entity ☒ Large Entity

Pursuant to 37 C.F.R. 41.20(b)(2), the fee for filing the Appeal Brief is:

☐ $250.00 (Small Entity)

☒ This fee was previously paid on May 19, 2006.

TOTAL FEE DUE:

Notice of Appeal Fee	$0.00 (previously paid on November 22, 2005)
Extension Fee (if any)	$0.00
Total Fee Due	$0.00

PETITION FOR EXTENSION. If any extension of time is necessary for the filing of this Appeal Brief, and such extension has not otherwise been requested, such an extension is hereby requested, and the Commissioner is authorized to charge necessary fees for such an extension to our Deposit Account No. 06-0916.

Please charge any additional required fees to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P.

Dated: October 20, 2006

By: _____

Venk Krishnamoorthy, Ph.D.
Reg. No. 52,490

Post Office Address (to which correspondence is to be sent)

Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P.
901 N.Y. Ave. N.W.
Washington, D.C. 20001-4413
(202) 408-4000

PATENT
Customer Number 22,852
Attorney Docket No. 07447.0061-00000
Xerox Ref. No.: D/AOA25

CERTIFICATE UNDER 37 CFR § 1.10 OF MAILING BY "EXPRESS MAIL"

EV758329616	October 20, 2006
USPS Express Mail Label Number	Date of Deposit

I hereby certify that this correspondence is being deposited with the United States Postal Services "Express Mail Post Office to Addressee" service under 37 CFR §1.10 on the date indicated above and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

By: Cindy Bagletto
Cindy Bagletto

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	
Eric GAUSSIER et al.	
Application No.: 09/982,236	Group Art Unit: 2161
Filed: October 19, 2001	Examiner: NGUYEN, Cam Linh T.
For: METHODS, SYSTEMS AND ARTICLES OF MANUFACTURE FOR SOFT HIERARCHICAL CLUSTERING OF CO-OCCURRING OBJECTS	Confirmation No.: 7611

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450.

Sir:

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

In response to the Office Action mailed on July 20, 2006, and pursuant to 37 C.F.R. 41.31, Appellants present the following Supplemental Appeal Brief and request reinstatement of the Appeal, which was originally filed on November 22, 2005.

The Notice of Appeal and the fee set forth in 37 C.F.R. § 41.37 were originally filed on November 22, 2005 along with a request for a pre-appeal brief conference. The final rejection was withdrawn and prosecution was reopened following the decision of the pre-appeal brief conference panel. The Examiner then issued another final rejection

on Dec. 21, 2005, which was appealed in the Appeal Brief filed May 19, 2006. In response to the Appeal Brief, Examiner withdrew the previous final rejection and entered new grounds of rejection in an Office Action mailed July 20, 2006 ("Office Action"). Appellants appeal Examiner's rejection of the claims and request that the Board of Appeals reverse in whole the rejections of claims 1 - 26 and order the allowance of these claims.

I. Real Party Interest

The real party in interest is Xerox Corporation.

II. Related Appeals and Interferences

There are no related appeals or interferences at this time.

III. Status of Claims

Claims 1-26 are pending and stand rejected. Appellants appeal the rejections of claims 1- 26.

IV. Status of Amendments

All amendments for this application have been entered.

V. Summary of Invention

The application describes methods, systems, and articles of manufacture for soft hierarchical clustering of objects based on a co-occurrence of object pairs. Clustering allows data to be hierarchically grouped (or clustered) based on its characteristics, so that objects, such as text data in documents that are similar to each other are placed in a common cluster in a hierarchy. In soft hierarchical clustering an object may be assigned to more than one cluster in a hierarchy as opposed to a hard assignment whereby an object is assigned to only one cluster in the hierarchy.

A modified Expectation-Maximization (EM) process is performed on object pairs reflecting documents and words, respectively, such that a given class of the objects ranges over all nodes of a topical hierarchy (as opposed to the leaves alone) and the assignment of a document to a topic may be based on any ancestor of the given class. Moreover, the assignment of a given document to any topic in the hierarchy may also be based on a particular (document, word) pair under consideration during the process. The modified EM process may be performed for every child class that is generated from an ancestor class until selected constraints associated with the topical hierarchy are met. A representation of the resultant hierarchy of topical clusters may be created and made available to entities that request the topics of the document collection. See e.g. pg. 4 lines 22-23, and pg. 5, lines 1-11.

The modified algorithm eliminates the reliance on leaf nodes alone and allows any set S_i to be explained by a combination of any leaves and/or ancestor nodes included in an induced hierarchy. That is, *i* objects may not be considered as blocks, but rather as pieces that may be assigned in a hierarchy based on any *j* co-occurring objects. In one configuration, a topical clustering application performed by a computer may assign parts of a document *i* to different nodes in an induced hierarchy for different words *j* included in the document *i*. See e.g. pg. 15, lines 10-20.

For example, the probability of observing any pair of co-occurring objects, such as documents and words (*i*, *j*), may be modeled by defining a variable $l_{r\alpha}$ (controls the assignment of documents to a hierarchy) such that it is dependent on the particular document and word pair (*i*, *j*) under consideration during a topical clustering process. In one configuration, the class α may range over all nodes in an induced hierarchy in order

to assign a document (*i* object) to any node in the hierarchy, not just leaves. Furthermore, by defining a class *v* as any ancestor of α in the hierarchy the nodes may be hierarchically organized. See e.g. pg. 15, lines 21-23, and pg. 16, lines 1-6.

Different *j* objects may be generated from different vertical paths of an induced hierarchy. That is, from paths in the hierarchy associated with non null values of $l_{i\alpha}$. Furthermore, because α may be any node in the hierarchy, the *i* objects may be assigned to different levels of the hierarchy. Accordingly, implementation of the model results in a pure soft hierarchical clustering of both *i* and *j* objects by eliminating any hard assignments of these objects. See, e.g., pg. 18, lines 10- 21.

The model may be implemented for a variety of applications, depending upon the meaning given to objects *i* and *j*. For example, it may be applied to document clustering based on topic detection. In such a configuration, *i* objects may represent documents and *j* objects may represent words included in the documents. Clusters or topics of documents may be represented by leaves and/or nodes of an induced hierarchy. The topics associated with the document collection may be obtained by interpreting any cluster as a topic defined by the word probability distributions, $p(j|v)$. The soft hierarchical model may take into account several properties when interpreting the clusters, such as: (1) a document may cover (or be explained by) several topics (soft assignment of *i* objects provided by the probability $p(i|\alpha)$); (2) a topic is best described by a set of words, which may belong to different topics due to polysemy (the property of a word to exhibit several different, but related meanings) and specialization (soft assignment of *j* objects provided by the probability $p(j|v)$); and (3) topics may be

hierarchically organized, which corresponds to the hierarchy induced over clusters. See, e.g., pg. 20, lines 25-30, and pg. 21, lines 1-11.

One or more conditions associated with a hierarchy that may be induced may allow a computer to determine when an induced hierarchy reaches a desired structure with respect to the clusters defined therein. For example, a condition may be defined that instructs a processor to stop locating co-occurring objects (*i*, *j*) in a document collection that is being clustered based on a predetermined number of leaves, and/or a level of the induced hierarchy. See, e.g., pg. 23, lines 1-11.

Pending independent claim 1 recites a method performed by a computer for clustering a plurality of documents in a structure comprised of a plurality of clusters hierarchically organized, wherein each document includes a plurality of words and is represented as a set of (document, word) pairs, the method comprising: accessing the document collection; performing a clustering process that creates a hierarchy of clusters that reflects a segregation of the documents in the collection based on the words included in the documents, wherein any document in the collection may be assigned to a first cluster in the hierarchy based on a first segment of the respective document, and the respective document may be assigned to a second cluster in the hierarchy based on a second segment of the respective document, wherein the first and second clusters are associated with different paths of the hierarchy; storing a representation of the hierarchy of clusters in a memory; and making the representation available to an entity in response to a request associated with the document collection. Claims 2 - 7 all ultimately depend from claim 1.

Pending independent claim 8 recites a method performed by a computer for determining topics of a document collection, the method comprising accessing the document collection, each document including a plurality of words and being represented as a set of (document, word) pairs; performing a clustering process including: creating a tree of nodes that represent topics associated with the document collection based on the words in the document collection, wherein any node in the tree may include a word that is shared by another node in the tree, and assigning fragments of one or more documents included in the document collection to multiple nodes in the tree based on the (document, word) pairs; storing a representation of the tree in a memory; and making the representation available for processing operations associated with the document collection. Claim 9 ultimately depends from claim 8.

Pending independent claim 10 recites a method performed by a processor for clustering data in a database, the method comprising: receiving a collection of documents, each document including a plurality of words and being represented as a set of (document, word) pairs; creating a first ancestor node reflecting a first topic based on words included in the collection of documents; creating descendant nodes from the first ancestor node, each descendant node reflecting descendant topics based on the first node, until a set of leaf nodes reflecting leaf topics are created. The step of creating descendant nodes includes assigning each document in the collection to a plurality of descendant and leaf nodes; and providing a set of topics associated with the collection of documents based on the created nodes and assignment of documents, wherein the descendant and leaf nodes may be created based on one or more words included in

more than one document in the collection of documents. Claim 11 ultimately depends from claim 10.

Pending independent claim 12 recites a method performed by a processor for clustering data in a database, the method comprising: receiving a collection of documents, each document including a plurality of words and being represented as a set of (document, word) pairs; creating a hierarchy of nodes based on the words in the collection of documents, each node reflecting a topic associated with the documents, wherein the hierarchy of nodes includes ancestor nodes, descendant nodes, and leaf nodes; assigning each document in the collection to a plurality of nodes in the hierarchy, wherein each document may be assigned to any of the ancestor, descendant, and leaf nodes; and providing a set of topic clusters associated with the collection of documents based on the created nodes and assignment of documents, wherein the hierarchy may include a plurality of nodes that are each created based on a same set of words included in the collection of documents.

Pending independent claim 13 recites a method performed by a computer for clustering data stored on a computer-readable medium, the method comprising: receiving a collection of data objects, represented as a set of (first data object, second data object) pairs; for each first data object: assigning the first data object to a first node in a hierarchy of nodes based on the second data objects included in the first data object, wherein the first node may be any node included in the hierarchy and wherein two or more nodes in the hierarchy may share the same second object; creating a final hierarchy of nodes arranged in clusters based on the assignment of the first data objects; storing a representation of the final hierarchy in a memory; and making the

representation of the final hierarchy available to an entity in response to a request associated with the collection of first data objects.

Pending independent claim 14 recites a method performed by a processor for clustering data in a database, the method comprising: receiving a request from a requesting entity to determine topics associated with a collection of documents, each document including a plurality of words and being represented as a set of (document, word) pairs; determining the topics associated with the collection of documents based on a hierarchy including a plurality of clusters, wherein each cluster reflects a topic and a document in the collection may be assigned to a set of clusters in the hierarchy based on different words included in the document, and wherein each cluster in the set may be associated with different paths in the hierarchy; storing a representation of the hierarchy in a memory; and making the representation available to the requesting entity.

Pending independent claim 15 recites a computer-implemented method for clustering a plurality of multi-word documents into a hierarchical data structure including a root node associated with a plurality of sub-nodes, wherein each sub-node is associated with a topic cluster based on the plurality of documents, the method comprising: retrieving a first document; associating the first document with a first topic cluster based on a first portion of the first document; associating the first document with a second topic cluster based on a second portion of the document; and providing a representation of topics associated with the plurality of multi-word documents based on the hierarchical data structure including the first and second topic clusters, wherein the first and second topic clusters are associated with a different sub-node. Claims 16 - 19 all ultimately depend from claim 15.

Pending independent claim 20 recites a computer-implemented method for clustering data reflecting users, represented as a set of (data, user) pairs, into a hierarchical data structure including a root node associated with a plurality of sub-nodes, wherein each sub-node represents an action that is performed on a document collection, comprising: accessing a user data collection reflecting a plurality of users who each perform at least one action on the document collection, wherein each action may be unique; performing a clustering process that creates the hierarchical data structure, wherein the clustering processing comprises: retrieving a first user data, associated with a first user, from the user data collection, associating the first user data with a first sub-node based on a first action performed by the first user on the document collection, and associating the first user data with a second sub-node provided the first user data is based on a second action, wherein the first and second sub-nodes are associated with different descendent paths of the hierarchical data structure; storing a representation of the hierarchical data structure in a memory; and making the representation available to an entity in response to a request associated with the user data collection. Claim 21 ultimately depends from claim 20.

Pending independent claim 22 recites a computer-implemented method for clustering a plurality of images based on text associated with the images, where each image is represented as a set of pairs (image, image feature) and (image, text feature), into a hierarchical data structure including a root node associated with a plurality of sub-nodes, wherein each sub-node represents a different topic, the method comprising: accessing an image collection; performing a clustering process that creates the hierarchical data structure, wherein the clustering processing comprises: associating a

first image with a first sub-node based on a first portion of text associated with the first image, and associating the first image with a second sub-node based on a second portion of text associated with the first image, wherein the first and second sub-nodes are associated with different descendant paths of the hierarchical data structure; storing a representation of the hierarchical data structure in a memory; and making the representation available to an entity in response to a request associated with the image collection.

Pending independent claim 23 recites a computer-implemented method for clustering customer purchases, represented as a set of (customer, purchase) pairs, into a hierarchical data structure including a root node associated with a plurality of sub-nodes, wherein each sub-node represents a group of customers who purchased the same type of product from one or more business entities, the method comprising: accessing information associated with a plurality of customers who purchased various types of products from a plurality of business entities; performing a clustering process that creates the hierarchical data structure, wherein the clustering processing comprises: associating a first customer with a first sub-node based on a first type of product purchased from a first business entity, and associating the first customer with a second sub-node provided the first customer is based on a second type of product that the first customer purchased from a second business entity, wherein the first and second sub-nodes are associated with different descendant paths of the hierarchical data structure; storing a representation of the hierarchical data structure in a memory; and making the representation available in response to a request associated with the customer data collection. Claims 24-26 all ultimately depend from claim 23.

VI. Grounds of Rejection to be Reviewed on Appeal

A. Whether claims 1- 26 should be rejected under 35 U.S.C. § 102 as unpatentable in light of U.S. Patent No. 6,742,003 B2 to Heckerman et. al. ("Heckerman")..

VII. Argument

As Appellants show below, Heckerman is directed to methods for reviewing web page access patterns of web users in order to optimize links between various web pages or to customize advertisements to the demographics of the users. Heckerman permits the visualization of clusters and cluster relationships between the web pages based on user access patterns where the data is represented as a collection of records, each containing values for various attributes. See, Heckerman, col. 1, lines 27 - 46. Heckerman does not disclose at least a method for clustering a plurality of documents in a collection, wherein each document is represented as a set of (document, word) pairs, and where a "document in the collection may be assigned to a first cluster in the hierarchy based on a first segment of the respective document, and the respective document may be assigned to a second cluster in the hierarchy based on a second segment of the respective document, wherein the first and second clusters are associated with different paths of the hierarchy," as recited in claim 1.

Figures 1A - 1D provide an overview of the methods outlined in Heckerman. Fig. 1B illustrates the results of the manual **classification** of a collection of records shown in Fig.1A. **Classification** techniques allow the manual grouping of the records of a collection into classes. Once classification has been performed, a new record may be automatically classified when it is added to the collection, as shown in Fig. 1C. In contrast, **clustering** techniques provide an automated process for analyzing the records

of the collection and identifying clusters of records that have similar attributes. Figure 1D illustrates the results of the clustering process performed in Heckerman on the collection of records shown in Fig.1A. *Id.*, col. 1, line 47 - col. 2, line 23. As stated in Heckerman, "the values stored in the column marked "CLUSTER" in FIG. 1D have been determined by the clustering algorithm." *Id.*, col. 2, lines 15 -18 and lines 27 - 32. The clustering method described in Heckerman assigns a record to only one of several clusters. *Id.*, col. 2, lines 30 - 32.

Other sections in Heckerman also reiterate that a record may belong to only one cluster.

> Clustering process 1510 automatically, and using a conventional clustering process, such as "EM" clustering, reads, as symbolized by lines 1503, data for the cases, in a dataset (population or collection) stored within case data 100 and automatically determines applicable **mutually exclusive** categories for these cases and then categorizes (classifies) each of those cases into those categories.

Heckerman, col. 25, lines 9 - 15 (emphasis added).

Further, the requirement that clusters contain mutually exclusive records underpins techniques outlined in Heckerman. For example, the technique to compute a discriminative score for cluster (group) c1 versus cluster (group) c2 given observation X=x, requires that clusters c1 and c2 are mutually exclusive. *Id.*, col. 32, lines 57 - 59. Other passages in Heckerman also teach that a record may only belong to one cluster.

> For each such user, database 1360 contains dataset 100 that contains a record for each such user along with predefined attributes (illustratively numbered 1 through j) for that user, and **the class** (**category or cluster**) to which that record is categorized. As noted, each such record together with all its attributes is commonly referred to as a "case". In addition, database 1360 also contains cluster data 1355 which specifies, e.g., clusters, segment and segment hierarchies.

Id., col. 21, lines 43 - 51 (emphasis added).

Moreover, as clearly indicated in Fig. 18 and the associated description, the sum of the percentages of cases in individual segments is equal to the total population. If a case belonged to multiple clusters (and therefore to multiple segments), then the sum of the percentage of cases in individual segments would exceed the total population.

> Segments are clusters of cases that exhibit similar behavior, such as users on a given site, and have similar properties, such as age or gender. A segment consists of a summary of the database records (cases) that belong to it.

Id., col. 21, lines 61 - 64.

> Display 1800 shows segment hierarchy 1810 in a left portion of the display. A user, such as a business manager or data analyst, by clicking on a down arrow displayed within hierarchy 1810 can expand a segment group to expose its constituent segments, as shown. Each segment and group are listed along with their corresponding percentages of an entire population. In that regard, segment 5 represents 10% of the entire population, segment group 6 represents 27% of the entire population, and so forth. As depicted, segment group 6 also contains segments 3 and 4.

Id., col. 22, lines 61 - 64.

As shown in Fig. 18, the individual percentages of the total number of cases associated with segments 5, 6, and 8, are 10%, 27%, and 63%, respectively, which add up to 100%, representing the total population.

Contrary to the Examiner's assertions (*See*, Office Action, on page 3, lines 12-21, citing Fig. 7; col. 2, lines 1-3; and col. 14, line 47 - col. 15, line 5), the scattered passages and drawings taken out of context do not disclose that "document in the collection may be assigned to a first cluster in the hierarchy based on a first segment of the respective document, and the respective document may be assigned to a second cluster in the hierarchy based on a second segment of the respective document, wherein the first and second clusters are associated with different paths of the hierarchy."

For example, the passages in col. 2, lines 1 -3 cited by the Examiner (*See, Id.*), relate to the **classification** of records by a web-master or data analyst, a process requiring human-intervention and not to the **clustering** of records. Heckerman clearly distinguishes the classification of records from clustering.

> **Classification techniques** allow a data analyst (e.g., web master) to group the records of a collection (dataset or population) into classes. That is, the data analyst reviews the attributes of each record, identifies classes, and then assigns each record to a class. ...
>
> However, certain records may have attributes that are similar to more than one class. Therefore, some classification techniques, and more generally some categorization techniques, assign a probability that each record is in each class. For example, record 1 may have a probability of 0.75 of being in class A, a probability of 0.1 of being in class B, and a probability of 0.15 of being in class C.

Heckerman, col. 1, line 57 - col. 2, line 8 (emphasis added).

On the other hand, Heckerman states, "**clustering** provides an automated process for analyzing the records of the collection and identifying clusters of records that have similar attributes." *Id.*, col. 2, lines 19 - 21. Therefore, contrary to the Examiner's opinion neither the cited passages, nor any other sections in Heckerman teach the assignment of a document to more than one cluster.

The Examiner also mischaracterizes Fig. 7 and its associated description. Fig. 7 illustrates a decision tree that allows a web master to determine the likelihood that a category of web page will be traversed when a user takes a specified course of action.

> Decision tree 700 contains nodes corresponding to attributes and arcs corresponding to values of that attribute. ...Thus, each node, except the root node, represents a setting of attribute values as indicated by the arcs in the path from that node to the root node. When a data analyst selects a node, the CV system displays a probability for each category that a record in that category will have the attribute settings that are represented by the path. For example, when the data analyst selects node 703 representing the attribute setting of accessing the "workshop" web page at least once and accessing the "intdev" web page twice, the CV system displays table

> 704. The table identifies the categories, the number of records in each category that matches those attribute settings, and the probabilities. For example, the first line "0 5 0.0039" indicates that category 0 has 5 records that match the attribute settings and that the probability for category 0 is 0.0039.

Id., col. 14, line 47 - col. 15, line 15.

Thus, contrary to the Examiner's assertion (*See,* Office Action, page 3, lines 12-21), FIG. 7 and the cited passages do not show a document that is assigned to a first and second cluster wherein the "first and second clusters are associated with different paths of the hierarchy" as recited in claim 1 of the application. Indeed, the clustering process in Heckerman automatically reads data for the cases, in a dataset and automatically determines applicable mutually exclusive categories for these cases and then categorizes each of those cases into those categories. Because categories contain mutually exclusive cases, a case cannot belong to more than one category. Claims 1, 19, 38, and 53 in Heckerman also clearly recite that the data records are classified based on the attribute / value pairs associated with each such record, into a plurality of "**mutually exclusive** first clusters" (emphasis added) further refuting the Examiner's opinion. Heckerman, col. 33, lines 50-53. Therefore, as taught in Heckerman, a data record in one cluster cannot belong to any other cluster.

For the reasons outlined above, Heckerman also does not teach at least the process of

> performing a clustering process including: creating a tree of nodes that represent topics associated with the document collection based on the words in the document collection, wherein any node in the tree may include a word that is shared by another node in the tree, and assigning fragments of one or more documents included in the document collection to multiple nodes in the tree based on the (document, word) pairs;

as recited in claim 8.

For the reasons outlined above, Heckerman also does not teach at least the process of creating descendant nodes, which includes

> assigning each document in the collection to a plurality of descendant and leaf nodes; and providing a set of topics associated with the collection of documents based on the created nodes and assignment of documents, wherein the descendant and leaf nodes may be created based on one or more words included in more than one document in the collection of documents

as recited in claim 10.

For the reasons outlined above, Heckerman also does not teach at least the process of

> assigning each document in the collection to a plurality of nodes in the hierarchy, wherein each document may be assigned to any of the ancestor, descendant, and leaf nodes; and providing a set of topic clusters associated with the collection of documents based on the created nodes and assignment of documents, wherein the hierarchy may include a plurality of nodes that are each created based on a same set of words included in the collection of documents

as recited in claim 12.

For the reasons outlined above, Heckerman also does not teach at least the process of

> assigning the first data object to a first node in a hierarchy of nodes based on the second data objects included in the first data object, wherein the first node may be any node included in the hierarchy and wherein two or more nodes in the hierarchy may share the same second object; creating a final hierarchy of nodes arranged in clusters based on the assignment of the first data objects

as recited in claim 13.

For the reasons outlined above, Heckerman also does not teach at least the process of

> determining the topics associated with the collection of documents based on a hierarchy including a plurality of clusters, wherein each cluster

reflects a topic and a document in the collection may be assigned to a set of clusters in the hierarchy based on different words included in the document, and wherein each cluster in the set may be associated with different paths in the hierarchy

as recited in claim 14.

For the reasons outlined above, Heckerman also does not teach at least the process of

providing a representation of topics associated with the plurality of multi-word documents based on the hierarchical data structure including the first and second topic clusters, wherein the first and second topic clusters are associated with a different sub-node

as recited in claim 15.

For the reasons outlined above, Heckerman also does not teach at least the process of

associating the first user data with a second sub-node provided the first user data is based on a second action, wherein the first and second sub-nodes are associated with different descendent paths of the hierarchical data structure

as recited in claim 20.

For the reasons outlined above, Heckerman also does not teach at least the process of

associating the first image with a second sub-node based on a second portion of text associated with the first image, wherein the first and second sub-nodes are associated with different descendant paths of the hierarchical data structure

as recited in claim 22.

For the reasons outlined above, Heckerman also does not teach at least the process of

associating the first customer with a second sub-node provided the first customer is based on a second type of product that the first customer

purchased from a second business entity, wherein the first and second sub-nodes are associated with different descendant paths of the hierarchical data structure

as recited in claim 23.

Therefore claims 1, 8, 10, 12, 13, 14, 15, 20, 22, and 23 are patentable over Heckerman.

Claims 2 - 7 depend from claim 1 and are patentable for at least the same reasons as is claim 1.

Claim 9 depends from claim 8 and is patentable for at least the same reasons as is claim 8.

Claim 11 depends from claim 10 and is patentable for at least the same reasons as is claim 10.

Claims 16 - 19 depend from claim 15 and are patentable for at least the same reasons as is claim 15.

Claim 21 depends from claim 20 and is patentable for at least the same reasons as is claim 20.

IX. Conclusion

A fundamental rule of patent examining procedure is that the burden is on the Examiner to establish that a reference teaches each and every element of the claims before any claim can be properly rejected under 35 U.S.C. § 102. The Examiner has failed to make such a showing in this case and failed to provide any support to show that the elements recited in the claims are taught by Heckerman.

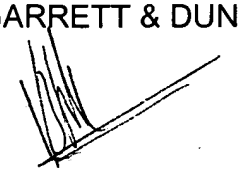
For the foregoing reasons, Appellants respectfully request reversal of all of the bases for rejection set forth in the Grounds of Rejection to be Reviewed on Appeal

section above (i.e., Section VI.A) and allowance of all pending claims.

To the extent any further extension of time under 37 C.F.R. § 1.136 is required to obtain entry of this Appeal Brief, such extension is hereby respectfully requested. Please grant any extensions of time required to enter this paper and charge any additional required fees to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P.



Dated: October 20, 2006

Venk Krishnamoorthy, Ph.D.
Reg. No. 52,490

Post Office Address (to which correspondence is to be sent)

Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P.
901 N.Y. Ave. N.W.
Washington, D.C. 20001-4413
(202) 408-4000

Claims Appendix

Pending claims on appeal:

1. A method performed by a computer for clustering a plurality of documents in a structure comprised of a plurality of clusters hierarchically organized, wherein each document includes a plurality of words and is represented as a set of (document, word) pairs, the method comprising:

accessing the document collection;

performing a clustering process that creates a hierarchy of clusters that reflects a segregation of the documents in the collection based on the words included in the documents, wherein any document in the collection may be assigned to a first cluster in the hierarchy based on a first segment of the respective document, and the respective document may be assigned to a second cluster in the hierarchy based on a second segment of the respective document, wherein the first and second clusters are associated with different paths of the hierarchy;

storing a representation of the hierarchy of clusters in a memory; and

making the representation available to an entity in response to a request associated with the document collection.

2. The method of claim 1, wherein performing a clustering process comprises:

assigning the document collection to a first class;

setting a probability parameter to an initial value; and

determining, for each document in the collection at the value of the parameter, a probability of an assignment of the document in the collection to a cluster in the hierarchy based on a word included in the document and the first class.

3. The method of claim 2, wherein the step of determining further comprises:

determining whether the first class has split into two child classes, wherein each child class reflects a cluster descendant from an initial cluster reflected by the first class; and

increasing the value of the parameter based on the determination whether the first class has split into two child classes.

4. The method of claim 3, further comprising:

repeating the step of determining, for each document in the collection at the value of the parameter, and the step of increasing the value of the parameter until the first class has split into two child classes.

5. The method of claim 4, further comprising:

performing the clustering process for each child class until each of the respective child class splits into two new child classes reflecting clusters descendant from the respective child class.

6. The method of claim 5, further comprising:

repeating the clustering process for each new child class such that a hierarchy of clusters is created, until a predetermined condition associated with the hierarchy is met.

7. The method of claim 6, wherein the predetermined condition is one of a maximum number of leaves associated with the hierarchy and depth level of the hierarchy.

8. A method performed by a computer for determining topics of a document collection, the method comprising:

accessing the document collection, each document including a plurality of words and being represented as a set of (document, word) pairs;

performing a clustering process including:

creating a tree of nodes that represent topics associated with the document collection based on the words in the document collection, wherein any node in the tree may include a word that is shared by another node in the tree, and

assigning fragments of one or more documents included in the document collection to multiple nodes in the tree based on the (document, word) pairs;

storing a representation of the tree in a memory; and

making the representation available for processing operations associated with the document collection.

9. The method of claim 8, wherein the step of assigning comprises:

associating a set of documents in the document collection with a first class reflecting all of the nodes in the tree, wherein the set of documents may include all or some of the documents in the collection;

defining a second class reflecting any ancestor node of a node in the first class;

determining, for each document in the set, a probability that different words included in a respective document co-occurs with the respective document in any node in the tree based on the first and second classes; and

assigning one or more fragments of any document in the set to any node in the tree based on the probability.

10. A method performed by a processor for clustering data in a database, the method comprising:

receiving a collection of documents, each document including a plurality of words and being represented as a set of (document, word) pairs;

creating a first ancestor node reflecting a first topic based on words included in the collection of documents;

creating descendant nodes from the first ancestor node, each descendant node reflecting descendant topics based on the first node, until a set of leaf nodes reflecting leaf topics are created,

wherein creating descendant nodes includes:

assigning each document in the collection to a plurality of descendant and leaf nodes; and

providing a set of topics associated with the collection of documents based on the created nodes and assignment of documents,

wherein the descendant and leaf nodes may be created based on one or more words included in more than one document in the collection of documents.

11. The method of claim 10, wherein the step of creating descendant nodes comprises:

selecting a first document in the collection;

defining a first class that includes all of the nodes;

defining a second class that may include any ancestor node of any node included in the first class; and

determining, for each document in the collection, a target word of an object pair including a target document and the target word such that the first document equals the target document in the object pair based on a probability associated with the first and second classes; and

assigning the first document to any ancestor, descendant, and leaf node based on the determining.

12. A method performed by a processor for clustering data in a database, the method comprising:

receiving a collection of documents, each document including a plurality of words and being represented as a set of (document, word) pairs;

creating a hierarchy of nodes based on the words in the collection of documents, each node reflecting a topic associated with the documents, wherein the hierarchy of nodes includes ancestor nodes, descendant nodes, and leaf nodes;

assigning each document in the collection to a plurality of nodes in the hierarchy, wherein each document may be assigned to any of the ancestor, descendant, and leaf nodes; and

providing a set of topic clusters associated with the collection of documents based on the created nodes and assignment of documents,

wherein the hierarchy may include a plurality of nodes that are each created based on a same set of words included in the collection of documents.

13. A method performed by a computer for clustering data stored on a computer-readable medium, the method comprising:

receiving a collection of data objects, represented as a set of (first data object, second data object) pairs;

for each first data object:

assigning the first data object to a first node in a hierarchy of nodes based on the second data objects included in the first data object, wherein the first node may be any node included in the hierarchy and wherein two or more nodes in the hierarchy may share the same second object;

creating a final hierarchy of nodes arranged in clusters based on the assignment of the first data objects;

storing a representation of the final hierarchy in a memory; and

making the representation of the final hierarchy available to an entity in response to a request associated with the collection of first data objects.

14. A method performed by a processor for clustering data in a database, the method comprising:

receiving a request from a requesting entity to determine topics associated with a collection of documents, each document including a plurality of words and being represented as a set of (document, word) pairs;

determining the topics associated with the collection of documents based on a hierarchy including a plurality of clusters, wherein each cluster reflects a topic and a document in the collection may be assigned to a set of clusters in the hierarchy based on different words included in the document, and wherein each cluster in the set may be associated with different paths in the hierarchy;

storing a representation of the hierarchy in a memory; and

making the representation available to the requesting entity.

15. A computer-implemented method for clustering a plurality of multi-word documents into a hierarchical data structure including a root node associated with a plurality of sub-nodes, wherein each sub-node is associated with a topic cluster based on the plurality of documents, the method comprising:

retrieving a first document;

associating the first document with a first topic cluster based on a first portion of the first document;

associating the first document with a second topic cluster based on a second portion of the document; and

providing a representation of topics associated with the plurality of multi-word documents based on the hierarchical data structure including the first and second topic clusters,

wherein the first and second topic clusters are associated with a different sub-node.

16. The method of claim 15, wherein the first and second portions contain at least one unique word.

17. The method of claim 15, wherein associating the first document with a first topic cluster comprises:

assigning the plurality of multi-word documents to a first class;

setting a probability parameter to an initial value; and

determining, for the first document at the value of the parameter, a probability of an assignment of the first document to the first topic cluster based on a word included in the first document and the first class.

18. The method of claim 15, wherein associating the first document with a second topic cluster comprises:

assigning the plurality of multi-word documents to a first class;

setting a probability parameter to an initial value; and

determining a probability of an assignment of the first document to the second topic cluster based on a word included in the first document and the first class.

19. The method of claim 15, wherein providing a representation comprises:

providing the representation after each document in the plurality of multi-word documents has been associated with to at least one topic cluster corresponding to a sub-node in the hierarchy, wherein any of the plurality of multi-word documents may be

associated to more than one topic cluster based on different portions of the respective document.

20. A computer-implemented method for clustering data reflecting users, represented as a set of (data, user) pairs, into a hierarchical data structure including a root node associated with a plurality of sub-nodes, wherein each sub-node represents an action that is performed on a document collection, comprising:

accessing a user data collection reflecting a plurality of users who each perform at least one action on the document collection, wherein each action may be unique;

performing a clustering process that creates the hierarchical data structure, wherein the clustering processing comprises:

retrieving a first user data, associated with a first user, from the user data collection,

associating the first user data with a first sub-node based on a first action performed by the first user on the document collection, and

associating the first user data with a second sub-node provided the first user data is based on a second action, wherein the first and second sub-nodes are associated with different descendent paths of the hierarchical data structure;

storing a representation of the hierarchical data structure in a memory; and

making the representation available to an entity in response to a request associated with the user data collection.

21. The method of claim 20, wherein each action in the one or more actions includes:

writing to, printing, and browsing the document collection.

22. A computer-implemented method for clustering a plurality of images based on text associated with the images, where each image is represented as a set of pairs (image, image feature) and (image, text feature), into a hierarchical data structure including a root node associated with a plurality of sub-nodes, wherein each sub-node represents a different topic, the method comprising:

accessing an image collection;

performing a clustering process that creates the hierarchical data structure, wherein the clustering processing comprises:

associating a first image with a first sub-node based on a first portion of text associated with the first image, and

associating the first image with a second sub-node based on a second portion of text associated with the first image, wherein the first and second sub-nodes are associated with different descendant paths of the hierarchical data structure;

storing a representation of the hierarchical data structure in a memory; and

making the representation available to an entity in response to a request associated with the image collection.

23. A computer-implemented method for clustering customer purchases, represented as a set of (customer, purchase) pairs, into a hierarchical data structure including a root node associated with a plurality of sub-nodes, wherein each sub-node represents a group of customers who purchased the same type of product from one or more business entities, the method comprising:

accessing information associated with a plurality of customers who purchased various types of products from a plurality of business entities;

performing a clustering process that creates the hierarchical data structure, wherein the clustering processing comprises:

associating a first customer with a first sub-node based on a first type of product purchased from a first business entity, and

associating the first customer with a second sub-node provided the first customer is based on a second type of product that the first customer purchased from a second business entity, wherein the first and second sub-nodes are associated with different descendant paths of the hierarchical data structure;

storing a representation of the hierarchical data structure in a memory; and

making the representation available in response to a request associated with the customer data collection.

24. The method of claim 1, wherein the representation defines the probability of a document as the product of the probability of the (document, word) pairs it contains.

25. The method claim 24, wherein the product is calculated after mixing the document-word pairs over the clusters.

26. The method claim 25, wherein mixing the (document, word) pairs over the clusters comprises a probability model of the form:

$$P(x) = \sum_{c} P(c)P(x \mid c)$$

wherein c is the group of clusters involved in the calculation, and x is a (document, word) pair.

Evidence Appendix

None.

Related Proceedings Appendix

None.